Cooperative Saliency-Based Pothole Detection and AR Rendering for Increased Situational Awareness

Gerasimos Arvanitis, Nikolaos Stagakis, Evangelia I. Zacharaki,
and Konstantinos Moustakas, *Senior Member, IEEE*

***Abstract*— Autonomous vehicles are expected to operate safely in real-life road conditions in the next years. Nevertheless, unanticipated events such as the existence of unexpected objects in the range of the road, can put safety at risk. The advancement of sensing and communication technologies and Internet of Things may facilitate the recognition of hazardous situations and information exchange in a cooperative driving scheme, providing new opportunities for the increase of collaborative situational awareness. Safe and unobtrusive visualization of the obtained information may nowadays be enabled through the adoption of novel Augmented Reality (AR) interfaces in the form of windshields. Motivated by these technological opportunities, we propose a saliency-based distributed, cooperative and rendering scheme for increasing the driver's situational awareness through (i) automated negative obstacle (potholes) detection, (ii) AR visualization and (iii) information sharing (upcoming potential dangers) with other connected vehicles or road infrastructure. An extensive evaluation study using a variety of real datasets for pothole detection showed that the proposed method provides favorable results and features compared to other recent and relevant approaches.**

***Index Terms*— Pothole detection, collaborative awareness, point cloud processing, augmented reality, CARLA, visualization, driver's safety.**

I. Introduction

INFORMATION-CENTRIC technologies have started to play a central role in the recent automotive industry boosting new research trends in semi or fully *Automated Driving Systems* (ADS). Autonomous vehicles, ranging from level 3 to level 5 of autonomy [1], are expected to operate safely in real-life road conditions, but the reality is that obstacles like potholes, bumps, and other unexpected objects are not uncommon in an everyday driving context. For this reason, the detection and identification of obstacles are imperative for reliable operation of autonomous vehicles [2]. Moreover, driver inattentiveness plays a major role in driving safety and is the culprit of road accidents around the world [3], [4], thus a lot of work has been devoted in the quantification of the abstract mechanics of human situational awareness [5]. Enhancing situational awareness is especially critical in the case of semi-autonomous cars, where the operator may be distracted by secondary activities, e.g. looking at the phone or reading a book. If the driver has to take over control, it is important to minimize the required reaction time. This can be achieved by monitoring and presenting to the driver the crucial information about the environment, thus keeping him/her aware of potentially hazardous situations. Inherent challenges include the need for unobtrusive information display, avoiding the effects of tunnel vision which could lead to actually overlooking critical information [6]. The problem of road pothole detection is commonly targeted using imaging (camera) data and computer vision techniques [7], [8], [9]. Although image-based techniques have achieved great success, one common drawback is that they are sensitive to motion blur and changes in lighting and/or even shadows [10]. Also, most techniques do not account for other passing vehicles [11]. This can make them unreliable in real use cases, which is a major weakness in problems involving human safety. In light of all this, the use of a 3D LiDAR (Light Detection and Ranging) sensor could provide more robust sensing capabilities for the analysis of potholes, in the same way that it is used to increase the accuracy of road's boundary detection [12], [13], [14]. On the other hand, a limitation of the LiDAR sensor is that, due to refraction and reflection, water appears as a black hole in the imagery calculated from LiDAR data [15], imposing additional challenges in the detection of potholes filled with water.

The purpose of this work is to increase the driver's situational awareness through automated cooperative obstacle detection, visualization and information sharing with other connected vehicles in a V2X (vehicle-to-everything) setting. To address the above issues, we developed a point cloud processing system that takes as input road environment data and classifies them into safe and potentially hazardous regions by identifying obstacles lying in the range of the road. We selected LiDAR as sensing modality for the surrounding environment due to its ability to retrieve depth information and its large range, making it suitable for driving environments. For more robust estimation, LiDAR data are fused with

Manuscript received 20 July 2022; revised 26 May 2023; accepted 20 October 2023. This work was supported by the European Union's Horizon 2020 Research and Innovation Program under Grant 101092875—DIDYMOS-XR: Digital DynaMic and responsible twinS for XR. The Associate Editor for this article was R. Arghandeh. *(Corresponding author: Gerasimos Arvanitis.)*

Gerasimos Arvanitis is with the Department of Electrical and Computer Engineering, University of Patras, 26504 Patras, Greece, and also with AviSense.ai Technologies, 26500 Patras, Greece (e-mail: arvanitis@ece.upatras.gr).

Nikolaos Stagakis, Evangelia I. Zacharaki, and Konstantinos Moustakas are with the Department of Electrical and Computer Engineering, University of Patras, 26504 Patras, Greece (e-mail: nick.stag@ece.upatras.gr; ezachar@upatras.gr; moustakas@ece.upatras.gr).

Digital Object Identifier 10.1109/TITS.2023.3327494

information on driving patterns, such as the steering angle of the wheels. For implementation and evaluation, we utilized the open-source CARLA simulator [16] including also a multi-agent system of vehicles, and we augmented it with our obstacle detection and tracking component. In this simulated environment, information sharing between agents is enabled, so that vehicles are notified about incoming obstacles even when there is no direct line-of-sight. Our method is capable of detecting static obstacles both negative (e.g., potholes) and positive (e.g., speed bumps or hazardous objects) within the range of the road. However, according to the literature the detection of negative obstacles poses more challenges, which have not been handled efficiently by the available methods so far [8], and is more specific to driving scenarios, in contrast to general object detection which has been extensively being studied in mobile robot navigation research.

To avoid any information visualization clutter, we propose the use of AR for visualizing critical information in the driver's field of view. AR rendering is based on classical perspective projection, where for each point (of the point cloud) the pixel coordinates in the image space of the AR interface are calculated through projection and a color is assigned indicating the object class. Interfaces that can be used for in-vehicle visualization include AR headset, Head-Up Display (HUD) [17], [18] or even the car's windshield with transparent display. The contributions of the proposed approach can be summarized as follows.

- Development of an obstacle detection module that takes into account the extraction of saliency maps from point clouds.
- Generation of data for randomized multi-ego connected vehicle in cooperative driving scenarios.
- Creation of realistic synthetic data of potholes that can be entered in the town maps of the CARLA simulator for the design of lifelike driving situations.
- AR visualization for point cloud projection registered on the scene images.
- Development of public and open access libraries with code for the aforementioned components[1],[2],[3], [4].

The rest of this paper is organized as follows. First we present previous works in related domains in Section II, and then describe in detail the proposed methodology in Sections III and IV. Section V follows with some experimental results in comparison with other state-of-the-art methods, while Section VI draws the conclusions and directions for future work.

II. Previous Work

In the following, we provide an overview of methodologies tackling the main challenges of the presented approach.

1) Negative Obstacle Detection: A major element that adds unpredictability in path planning for self-driving cars are obstacles in the road. Negative obstacles can appear in the form of objects beyond the surface of the road, or cracks and holes in paved areas. There has been major work on obstacle detection, raging from real-time implementations [7], to offline schemes that act as automated informants to the authorities responsible for maintenance [19], or as efficient unsupervised techniques for pothole detection [20]. Most of the existing works implement a broad spectrum of computer vision and/or machine learning techniques to analyze imaging information [7]. The methods differ mainly on the utilized features and classifiers for obstacle representation and recognition. In respect to performance, a direct comparison of methods is not feasible because most works are evaluated on their own (simulated) data. In fact, there is lack or restricted access to a common benchmark dataset with potholes and obstacles, that can be used for comparison.

Asad et al. [21] explored the potential of deep learning models (YOLO family and SSD-mobilenetv2) for real-time pothole detection leading towards the deployment on edge devices. However, they used only images, and despite the good detection results, the visualization of the information was considered as disturbing for real-case implementation.

Jenkins and Young [22] discussed the design of a system to alert motorcyclists for different hazard categories including roadway hazards (e.g., potholes, other vehicles or pedestrians, roadway debris, uneven surfaces), in a manner that facilitates direct perception and action to appropriately respond to such hazards and reduce the risk of accidents and injuries. However, no real-time data processing is performed, and analysis is based on data that have been already stored in the cloud.

Heo et al. [23] proposed a 2D pothole risk assessment standard to visually indicate risk signals to the driver by comparing the size of the pothole detected using the developed model with the size of the tire contact patch area. The authors conclude that if a risk assessment method could be used in real-time, it may not only be useful for road maintenance but also for detecting large potholes that are not recognizable by drivers in driving situations.

Other methods focus on road cracks detection from high resolution cameras on smartphones. Since such data are more easily available, those methods can bypass the extraction of hand-crafted features and utilize deep architectures, such as convolutional neural networks [24], [25], [26], [27]. However, in the case of dense traffic situations and poor lighting conditions, techniques utilizing images from smartphone camera are less effective. In contrast to computer vision techniques which exploit texture information from images, 3D point cloud processing techniques exploit the object's geometrical properties [15], [28], [29]. Bosurgi et al. [29] identify potholes in road sections by estimating area, perimeter and depth information from 3D data of pavement surfaces. Chen et al. [15] propose a framework for obstacle detection using the pitch and rotation angles of a LiDAR sensor to create a 2D image-like plane where the unordered set of points (from the point cloud) are projected. From this "LiDAR-imagery" a 2D histogram is extracted and used to find the road plane. If an adequate part of the road, in front of the vehicle is flat, those points form a straight line in the histogram representation, and anything above the line can be classified as a positive obstacle (points

[1] https://github.com/Stagakis/saliency-from-pointcloud

[2] https://github.com/Stagakis/carla-data-generation

[3] https://github.com/Stagakis/roadpatch-with-pothole-generator

[4] https://github.com/Stagakis/carlapclprocessing

higher than the road plane), while points below the line as a negative obstacle (points lower than the road plane). Moreover, since water bodies cannot be detected by LiDAR due to refraction and reflection, the authors propose a technique to detect potholes filled with water by scanning the image for large areas of missing data.

Gu et al. [28] improves the aforementioned method by projecting the points on the camera plane and interpolating the depth values of the projected points to receive a depth image. They use both horizontal and vertical histograms to coarsely detect the road area and refine it respectively. Although they state their method as sensor fusion between the monocular camera and LiDAR, they do not utilize the color values of the camera images. Both works [15] and [28] use the KITTI dataset as a benchmark and achieve great results, comparable to machine learning methods.

Other techniques for pothole detection may include laser scanning, ground penetrating radar, ultrasonic sensor, as well as multi-sensor fusion, especially concerning fusion with imaging information. An extensive review of such techniques falls beyond the scope of this article. However, an interested reader may be referred to the survey in [30].

2) Point Cloud Saliency: One of main challenges in techniques utilizing point clouds is the inherent noise and the increased computational cost due to the unordered data structure of point clouds. To address such challenges, saliency map extraction has been proposed as a powerful step in point cloud processing to reduce noise and data dimensionality, leading to more robust solutions and computational efficiency [31], [32]. Yet, the use of local saliency in pothole detection has not been sufficiently examined. Saliency maps were constructed from point clouds obtained from Mobile Laser Scanning (MLS) in [33] for road crack detection. MLS point clouds contain spatial information (i.e., Euclidean coordinates) and intensity information, and thus the extracted features could leverage both height and intensity information. Feature saliency was estimated by calculating the distances from the normal of each point to the principal normal of the input point clouds. In a similar setting, Wang et al. [34] extracted saliency maps in MLS point clouds by projecting the distance of each point's normal vector to the point cloud's dominant normal vector into a hyperbolic tangent function space.

3) Cooperative Driving: While significant advances have been made for single-agent perception, many applications require multiple sensing agents and cross-agent communication for more accurate results. Objects, captured by the single-agent's sensor devices, may be heavily occluded or far away from the sensors' view, resulting in sparse observations. Nevertheless, failing to detect and predict the accurate position or moving intention of these occluded or "hard-to-see" objects might have harmful consequences in safety-critical situations, and especially if the reaction time is very narrow [35]. The development of multi-agent solutions can lead to collaborative perception and, through information sharing, may improve the driving performance and experiences, providing endless possibilities for safe driving.

Recently, cooperative autonomous driving has been considered as a possible solution to improve the performance and safety of autonomous vehicles [36]. Cooperative perception for 3D object detection can be performed via early or late fusion of information, i.e., combination of multiple sensing points of view or fusion of object detection results, respectively.Both fusion approaches can extend the perception of the sensing system, however, only the early fusion approach can actually exploit complementary information. A major challenge that arises regarding cooperative perception is how to effectively merge sensors' data received from different vehicles to obtain a precise and comprehensive perception outcome. Additionally, despite the attention that cooperative driving has attracted recently, the absence of a suitable open dataset for benchmarking algorithms has made it difficult to develop and assess cooperative perception technologies.

Xu et al. [37] presented the first open dataset and used it to benchmark fusion strategies for V2V (vehicle-to-vehicle) perception. They also plan to extend the dataset with more tasks as well as sensor suites and investigate more multimodal sensor fusion methods in the V2V and V2I (vehicle-to-infrastructure) settings. Arnold et al. [36] proposed a system that produces a perception of complex road segments (e.g., complex T-junctions and roundabouts) using a network of roadside infrastructure sensors with fixed positions. Chen et al. [38] studied the raw-data level cooperative perception for enhancing the detection ability of self-driving systems. They fuse the sensor data collected from different positions and angles of connected vehicles, relying on LiDAR 3D point clouds. Liu et al. [39] addressed the collaborative perception problem, where one agent is required to perform a perception task and can communicate and share information with other agents on the same task.

Chen et al. [40] proposed a point cloud feature-based cooperative perception framework for connected autonomous vehicles to increase object detection precision. The features are selected to be rich enough for the training process, and at the same time have an intrinsically small size to achieve real-time edge computing. Guo et al. [41] proposed a cooperative fusion method to combine spatial feature maps for achieving a higher 3D object detection performance. Yuan et al. [42] proposed a 3D keypoints feature fusion scheme for cooperative driving detection to remedy the problem of low bounding box localization accuracy. Fang et al. [43] presented an iterated split covariance intersection filter-based cooperative localization strategy with a decentralized framework. In addition, they adopted a point cloud registration method to obtain the relative pose estimation using mutually shared information from neighbour vehicles. Kim and Liu [44] presented the concept of cooperative autonomous driving using mirror neuron-inspired intention awareness and cooperative perception, providing information on the upcoming traffic situations ahead, even beyond line-of-sight and field-of-view.

4) Situational Awareness and AR Infotainment: In the case of semi-autonomous vehicles, where the operator/driver may be asked to take manual control of the car at any moment, it is of great importance [45] to implement notification paradigms that direct the operator's, possibly reduced, attention to the event that triggered the take-over request [46], [47]. Recently, the automotive industry started to invest funds and efforts into

AR technology and its integration with In-Vehicle Information Systems (IVIS) for intuitive and non-intrusive information display to the driver.

The design of AR in-vehicle systems for infotainment is a challenging task. Rao et al. [48] performed an analysis of design methods on different use cases aiming to identify the difficulties in implementation aspects. Despite the vast amount of requirements for these systems to work reliably, such as latency, bandwidth, weather conditions etc, they concluded that the integration of augmented reality in vehicles will help drivers navigate their environment better, and thus will be more widely adopted.

While IVIS existing in many modern vehicles with touch Liquid-Crystal Display (LCD) displays and voice commands may seem to offer most of the utilities of an AR infotainment system, they may actually be distracting to the driver. Strayer et al. [49] showed in a recent study that some IVIS require a high cognitive demand or complex command sequences to be handled, and this can in turn lower the awareness of the operator. This is perpetuated by the fact that most IVIS are placed on the dashboard and usually demand their operation to avert (even momentarily) the driver's gaze from the road. In contrast, AR HUDs perform information rendering on top of the environment and thus the driver does not need to share focus in multiple locations.

The distraction potential of AR HUDs was assessed by Kim and Gabbard [50]. An AR-enabled windshield was used in a simulated environment with a real-life driving video feed to test various methods of pedestrian visualization. The gaze behavior and cognitive processes were measured and it was found that the visual and cognitive distraction potential of AR depends on the perceptual forms of graphical elements presented on the displays. Specifically, in some cases visualizations, e.g., in the form of a "virtual transparent shadow" indicating the pedestrian's anticipated path, improved the driver's attention without degrading awareness of other objects or scene elements. On the other hand, the use of bounding boxes localizing pedestrians showed to have negative effects, because this approach either overloaded (visually) the scene or degraded the driver's attention on other – not highlighted but possibly critical – scene elements. These outcomes indicate that, while the potential of AR for improving situational awareness is tangible, a lot of attention must be paid for the AR design to not end up cluttering and obstructing the driver's attention.

Kettle and Lee [51] reviewed the AR visualizations for in-vehicle vehicle-driver communication, regarding factors like the display mode (e.g., windshield, simulated, HUB, etc.), the display design (e.g., bounding boxes, warning symbols, arrows, etc.) and the display information (e.g., hazard detection, pedestrian detection, vehicle detection, road signs, etc.). They concluded that there are many benefits of implementing AR interfaces, and such interfaces have the potential to improve driving performance through braking and takeover responses. The research on augmented reality displays on windshields for improving driver awareness also extends to fully Autonomous Vehicles (AV). Such informative human-machine interfaces may help to form a mental model of the vehicle's sensory and planning system, thereby enhancing trust in AV, which is currently quite low in the general public [52], [53], [54]. Lindemann et al. [55] conducted a user study on urban environments for evaluating the situational awareness of the driver in various scenarios. They found that their explanatory windshield display had positive results and improved the operator's trust. Yontem et al. [56] also designed an AR windshield interface targeting future vehicles. Their main focus was also to increase driver awareness by presenting graphical cues in a non-intrusive way based on a human-centric design and taking into account the human peripheral vision.

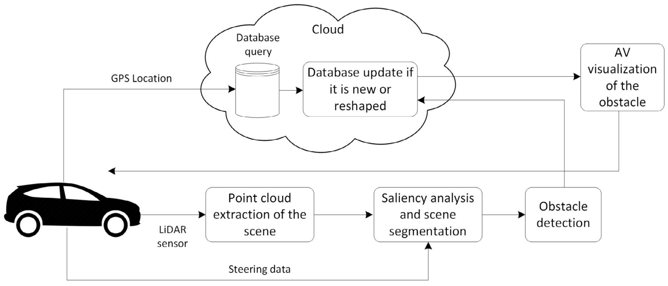

Cloud
Database query
Database update if it is new or reshaped
GPS Location
AV visualization of the obstacle
LiDAR sensor
Point cloud extraction of the scene
Saliency analysis and scene segmentation
Obstacle detection
Steering data

Fig. 1. Schematic diagram of the proposed methodology.

While the above methods provide essential feedback on the assessment of such interfaces' design, a significant limitation is that most studies were based on basic or non-interactive simulations, with the steering wheel and pedals not influencing the simulated environment and thus restricting the feeling of immersiveness of the simulations during the evaluation study. A more realistic, experimental study on the benefits of AR in driver's behavior was performed by Kim et al. [57] outdoors in a parking lot. It focused on pedestrian collision warning based on visual depth cues delivered in a conformal manner through a monocular display seated above the dashboard, or a volumetric display providing binocular disparity. A limitation of this study, which we address through our AR visualization component (subsection IV-A of section IV), is the limited field of view of the display used in the experiments, potentially creating a tunneling effect of the human vision.

III. Obstacle Detection

This section presents the proposed methodology on obstacle detection and is followed by section IV on visualization and communication aspects. The main components of the methodology are illustrated in the schematic diagram in Fig. 1 and can be encapsulated in the next steps:

- *Extraction of saliency map*: A saliency value is estimated for any point of the point cloud scene based on its local geometry, as well as the local geometry of its neighboring points.
- *Scene segmentation*: The estimated saliency map is then used as a feature to segment the point cloud into areas characterizing (i) the safe area of the road, (ii) be-aware or dangerous areas within the range of the road, and (iii) areas out of the range of the road.
- *Static object recognition*: Static objects (i.e., potholes and bumps) can be identified and their point coordinates are

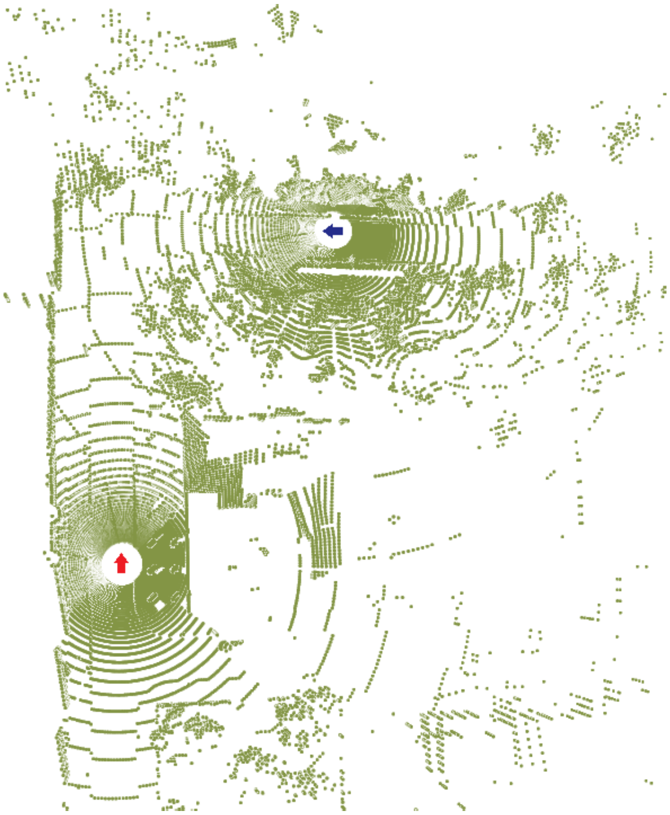


Fig. 2. Point cloud map of both two vehicles (ego1 and ego2).

stored and then used for the AR-based visualization and communication to other nearby vehicles.

In this work, we assume the existence of two or more vehicles (referred as *ego1* and *ego2* vehicles in this paper) that are moving on the same map of a town but not necessarily at the same time, i.e., they are in spatial proximity but possibly not in temporal proximity. Fig. 2 presents an example of two registered point clouds, as received by the LiDAR devices of *ego1* and *ego2* vehicles, showing also their starting points (in arrows). We would like to mention here that all the following analysis is applied to each vehicle separately.

A. Notations

Before presenting details on the individual steps, we provide here the necessary definitions and notations. The input data constitute a sequence of point clouds $\mathbf{P}_i$, $i = 1, \ldots, l$ that represents a set of l consecutive frames acquired by a LiDAR device. Each point cloud $\mathbf{P}_i$ consists of m_i vertices $\mathbf{v}$, where the value of m_i may be different from frame to frame. The j-th vertex ($\mathbf{v}_j$) of a point cloud $\mathbf{P}_i$ is represented by the Cartesian coordinates, denoted $\mathbf{v}_j = \left[x_j,\ y_j,\ z_j\right]^T,\ \forall\ j = 1, \cdots, m_i$, where the index i of the point cloud is omitted for simplification. Thus, all the vertices can be represented as a matrix $\mathbf{V} = \left[\mathbf{v}_1,\ \mathbf{v}_2,\ \cdots, \mathbf{v}_{m_i}\right] \in \mathbb{R}^{3\times m_i}$. Let's also denote with Ψ_j^k the set of the indices of the k nearest neighbors of point j. For a face f defined by three vertices $(\mathbf{v}_{j1}, \mathbf{v}_{j2}, \mathbf{v}_{j3})$, the outward unit face normal $\mathbf{n}_f$ is calculated by the following equation:

$$\mathbf{n}_f = \frac{\left(\mathbf{v}_{j2} - \mathbf{v}_{j1}\right) \times \left(\mathbf{v}_{j3} - \mathbf{v}_{j1}\right)}{\left\|\left(\mathbf{v}_{j2} - \mathbf{v}_{j1}\right) \times \left(\mathbf{v}_{j3} - \mathbf{v}_{j1}\right)\right\|} \tag{1}$$

The point normal $\mathbf{n}_j$, representing the normal of each point separately, is calculated as:

$$\mathbf{n}_j = \frac{\sum_{\forall \mathbf{n}_f \in \Psi_j^k} \mathbf{n}_f}{|\Psi_j^k|} \tag{2}$$

B. Saliency Map Estimation of the Point Cloud Scene

The purpose of this step is to calculate a metric of saliency for each vertex of a point cloud. Assuming point clouds without context information, saliency characterizes the geometric properties in a local neighborhood of points, i.e., high saliency values represent more perceptually prominent vertices which usually correspond to sharp corners (high-frequency spatial information). On the opposite, the geometrically least important points are those that lie in flat areas.

For the estimation of the saliency map, we implemented and modified the fusion technique presented in [32]. Instead of using guided normals of centroids, as in the original version [32], we now utilize normals for the points. This was performed to accelerate computations. Since the number of faces is usually approximately twice the number of vertices, the point normals are almost half the number of the centroid normals. For the sake of completeness, we present here our approach for the estimation of the saliency map of a point cloud scene, utilizing point normals.

Our fusion technique combines geometric saliency ($s^{(1)}$) with spectral saliency ($s^{(2)}$) features. The unique characteristics of each of these saliency features make the methodology more robust to point clouds acquired under real conditions, thereby being potentially affected by noise and outliers. The method processes each frame independently without examining past temporal information. Thus, as the methodology is applied for each point cloud in the sequence independently, for simplicity we omit the index i (indicating the frame number) from now on in the equations.

For a point cloud $\mathbf{P}$ with m vertices, a matrix $\mathbf{E} \in \mathbb{R}^{3m\times(k+1)}$ is constructed which includes in the first column the m point normals ($\mathbf{n}_j = [n_{j_x}, n_{j_y}, n_{j_z}]^T$) of each vertex j, $j = 1, \cdots, m$, respectively, and in the subsequent k columns the point normals of the k nearest neighbors of vertex j (i.e. $\mathbf{n}_{j\kappa} \in \Psi_j^k$). The salient features extracted by this approach capture global information since the matrix $\mathbf{E}$ is constructed using the point normals of the whole scene.

In order to exploit the geometrical coherence between neighboring normals, we apply Robust Principal Component Analysis (RPCA) to decompose the matrix $\mathbf{E}$ into a low-rank matrix $\mathbf{L} \in \mathbb{R}^{3m\times(k+1)}$ and a sparse matrix $\mathbf{S} \in \mathbb{R}^{3m\times(k+1)}$, as described in the appendix A. The matrix $\mathbf{L}$ consists of the low-rank values $\bar{\mathbf{n}}$ of the point normals $\mathbf{n}$, while the matrix $\mathbf{S}$ consists of the corresponding sparse values represented as $\dot{\mathbf{n}}$. The values of this matrix are zero (or to be more specific nearly zero) if the row (representing a neighboring patch of points) corresponds to point normals with very similar values, i.e., the vertex lies in a flat area, and very large values if the row corresponds to point normals with big dissimilarity (i.e., the vertex lies in a very sharp corner). The fact that most of the local patches Ψ_j^k of a 3D surface are piecewise flat confirms that the matrix $\dot{\mathbf{S}}$ can be considered a sparse matrix.

$$\mathbf{S} = \begin{bmatrix} \dot{\mathbf{n}}_1 & \dot{\mathbf{n}}_{11} & \dot{\mathbf{n}}_{12} & \dots & \dot{\mathbf{n}}_{1k} \\ \dot{\mathbf{n}}_2 & \dot{\mathbf{n}}_{21} & \dot{\mathbf{n}}_{22} & \dots & \dot{\mathbf{n}}_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \dot{\mathbf{n}}_m & \dot{\mathbf{n}}_{m1} & \dot{\mathbf{n}}_{m2} & \dots & \dot{\mathbf{n}}_{mk} \end{bmatrix} \tag{3}$$

In other words, sparsity of the matrix is assumed because piecewise flat areas are the most dominant geometrical pattern in a 3D surface.

1) Estimation of the Geometrical Saliency (global approach): As the similarity of normals between neighboring points is a measure of geometrical coherence of the local neighborhood, we estimate the sparsity of the dissimilarity of normals and use it as a feature for geometrical saliency, $s^{(1)}$. Low values of the sparse matrix indicate that the normals of the point and its neighbors are similar (low-rank). This means that if all points in a neighborhood have similar geometrical characteristics, the respective patch represents a flat area. On the opposite, high dissimilarity indicates that the surface has an irregular shape. For a point $\mathbf{v}_j$ the geometric saliency feature, $s_j^{(1)}$, is estimated by the values of the first column of the sparse matrix $\mathbf{S}$ according to:

$$s_j^{(1)} = ||\dot{\mathbf{n}}_j||^2 = \sqrt{\dot{n}_{j_x}^2 + \dot{n}_{j_y}^2 + \dot{n}_{j_z}^2} \quad \forall\ j = 1, \cdots, m \tag{4}$$

where $\dot{n}_{j_x}$ denotes the scalar value of the x coordinate, of the $[3 \cdot (j-1)+1]^{th}$ row, of the 1^{st} column of the $\mathbf{S}$ matrix.

2) Estimation of the Spectral Saliency (local approach): For the estimation of the spectral-based saliency, $s_j^{(2)}$, for a vertex j of the point cloud, we use the submatrix $\mathbf{E}_j \in \mathbb{R}^{3\times(k+1)}$, that includes the 3 corresponding rows of the matrix $\mathbf{E}$:

$$\mathbf{E}_j = \begin{bmatrix} n_{j_x} & n_{j_{x1}} & n_{j_{x2}} & \dots & n_{j_{xk}} \\ n_{j_y} & n_{j_{y1}} & n_{j_{y2}} & \dots & n_{j_{yk}} \\ n_{j_z} & n_{j_{z1}} & n_{j_{z2}} & \dots & n_{j_{zk}} \end{bmatrix}, \forall\ j = 1, \cdots, m \tag{5}$$

In other words, each submatrix $\mathbf{E}_j$, which is a subset of the global matrix $\mathbf{E}_i$, consists of the point normals of a local neighborhood of the vertex $\mathbf{v}_j$. Then, for each one of these local matrices $\mathbf{E}_j$, the covariance matrix $\mathbf{R}_j \in \mathbb{R}^{3\times 3}$ is calculated:

$$\mathbf{R}_j = \mathbf{E}_j \mathbf{E}_j^T \tag{6}$$

Next, the calculated matrix $\mathbf{R}_j$ is decomposed into a matrix $\mathbf{U}$ consisting of the eigenvectors and a diagonal matrix $\Lambda = \text{diag}(\lambda_{j1}, \lambda_{j2}, \lambda_{j3})$ consisting of the corresponding eigenvalues, i.e., $[\mathbf{U} \quad \Lambda] = \text{eig}(\mathbf{R}_j)$, where eig(.) represents the eigendecomposition operation.

Finally, the spectral saliency of each vertex is calculated by the inverse l^2-norm of the corresponding eigenvalues:

$$s_j^{(2)} = \frac{1}{\sqrt{\lambda_{j1}^2 + \lambda_{j2}^2 + \lambda_{j3}^2}} \quad \forall\ j = 1, \cdots, m \tag{7}$$

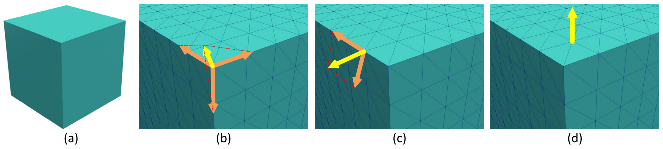


Fig. 3. (a) Cube model, (b) corner ($\lambda_{i1} \cong \lambda_{i2} \cong \lambda_{i3}$), (c) edge ($\lambda_{i1} \cong \lambda_{i2} > \lambda_{i3}$), (d) flat area ($\lambda_{i1} > \lambda_{i2} \cong \lambda_{i3}$).

Eq. (7) indicates that large values of the term $\sqrt{\lambda_{i1}^2 + \lambda_{i2}^2 + \lambda_{i3}^2}$ correspond to small saliency features implying that the centroid lies in a flat area, while small values of the eigenvalues' norm correspond to large saliency, characterizing the specific centroid as a discriminative point.

This can be easily justified by the fact that a point normal lying on a flat area is represented by one dominant eigenvector, the corresponding eigenvalue of which has a very large value (especially, considering that it is squared). On the other hand, the point normal of a vertex lying on a corner is represented by three eigenvectors, that correspond to eigenvalues with small and almost equal amplitude, as shown in Fig. 3.

3) Normalization and Fusion of Local and Global Saliency: Finally, we linearly scale the values of the geometric ($s^{(1)}$) and spectral ($s^{(2)}$) saliency in the range of [0-1] and combine them through weighted averaging according to:

$$s_j = \frac{w_1 \bar{s_j}^{(1)} + w_2 \bar{s_j}^{(2)}}{w_1 + w_2} \quad \forall\ j = 1, \cdots, m_i \tag{8}$$

where $\bar{s}^{(1)}$ and $\bar{s}^{(2)}$ denote the normalized geometric and spectral saliency features, and w_1 and w_2 the corresponding weights. We note here that we used equal weights ($w_1 = w_2 = 1$) in all of our experiments, however, the weights can be tuned to emphasize the local or global saliency descriptors, respectively.

The proposed method has shown to be robust [31], [32], even for complex surfaces with different geometrical characteristics and patterns, since it exploits spectral properties (i.e., sensitivity in the variation of neighboring normals) and geometrical characteristics (i.e., sparsity of intense prominent spatial features). An example of the visualization of the saliency map, as applied to the point cloud of a scene shown in Fig. 4 (a), is presented in Fig. 4 (b).

C. Scene Segmentation for the Identification of On-Road Obstacles

The saliency map of each frame is used to categorize different regions of the scene. For illustration purposes the regions are visualized in different colors:

- Blue: The safe area of the road beyond the view of the driver.
- Yellow: Be-aware areas representing negative obstacles.
- Cyan: Hazardous areas in the range of the road representing positive obstacles.
- Purple: Dangerous areas outside of the range of the road.
- Red: Recognized obstacles in the range of the road (e.g., potholes).

To define the vehicle's moving direction steering data are used received by internal sensors of the vehicle. The direction

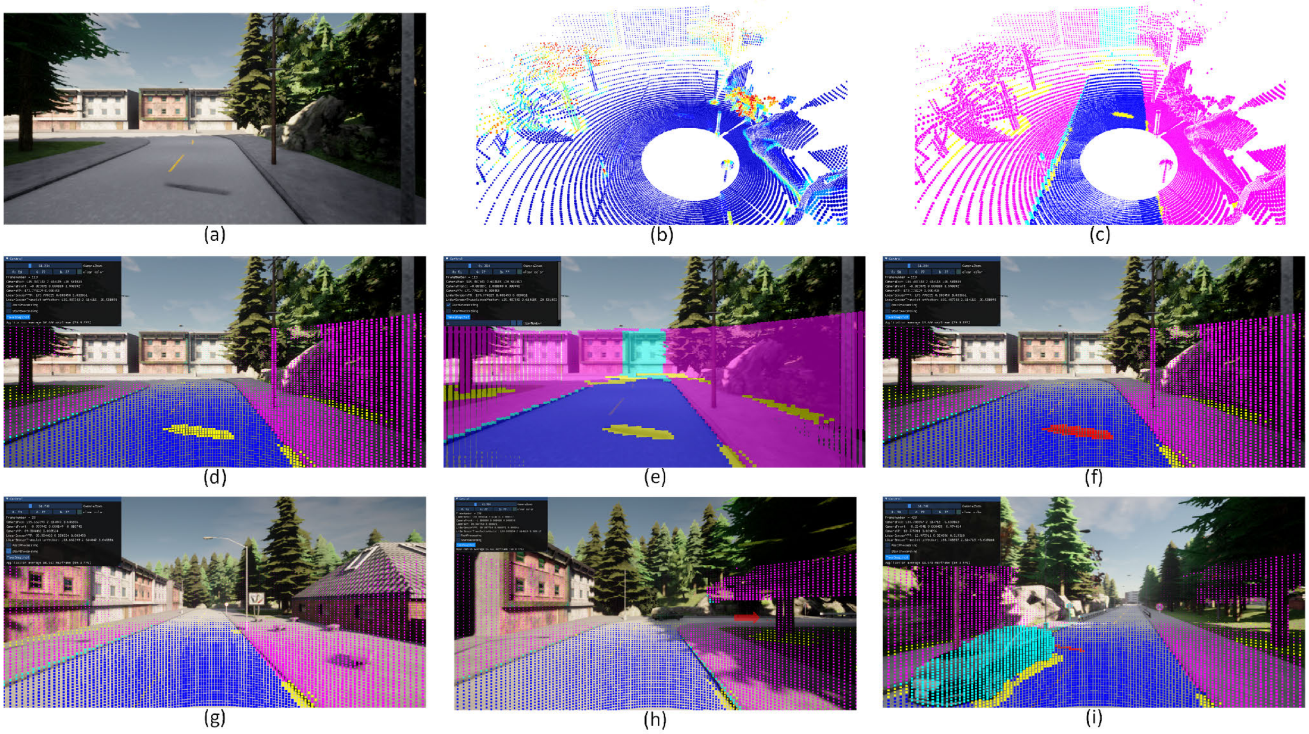


Fig. 4. (a) Image from the camera of the vehicle, the texture of a pothole is also apparent, (b) Extracted saliency map of the same road scene, (c) Segmentation of the point cloud scene based on the saliency map, (d) Example of segmentation of the point cloud projected to the AR interface (in the view of ego1), (e) Perspective projection of the point cloud vertices to the AR interface and image filling, (f) Pothole recognition (highlighted in red color) and AR visualization of the corresponding information (in the view of ego1), (g) AR projection of the point cloud vertices to the scene image that depicts the starting point of view of the ego2 vehicle, (h) Early warning of upcoming pothole to inform ego2, (i) Pothole recognition and visualization (in the view of ego2).

of the vehicle specifies which part of the scene in the field of view is in front of the vehicle and is used as parameter, in addition to saliency mapping, for the segmentation of the point cloud. The more critical regions are the ones that lie within the limits of the road. A segmentation example is illustrated in Fig. 4 (c).

D. Data Simulations

For evaluation of our methodology, we created a rich dataset using CARLA, an open-source autonomous driving simulator [16]. CARLA is based on a server-client system, in which the server is responsible for running the simulations including the calculation of physics, weather conditions, collision detection and sensor readings. It operates on the OpenDRIVE specification [58] for defining junctions, traffic lights, etc, and is used by CARLA for simulating independent agents, such as other cars and pedestrians. This makes CARLA ideal for creating complex scenarios and realistic driving conditions for our tests.

The server running the simulations is powered by Unreal Engine. Clients can connect and request changes to almost any element in the world being essential for the creation of scenarios. They also receive sensor data and manage input to the vehicle controlled by the user. CARLA supports a wide range of sensor suites with extensive configurability to its intrinsic parameters. In our work, we use a LiDAR sensor on top of the vehicle and a monocular RGB camera, placed in the front part of the car, for simulated data collection. By placing these sensors in an autonomous car and initiating its navigation in the virtual environment, we were able to create a very large dataset for evaluating our algorithms. In the future, we plan to assess the AR visualization effectiveness, with respect to reaction time and awareness increase, in a real environment with a driver manually controlling a vehicle.

Contributions in CARLA simulator: Due to lack of benchmark point clouds datasets representing real road scenes with obstacles (potholes and bumps), we used the CARLA simulator to create obstacle-free environment data, in which we subsequently introduced simulated obstacles. Specifically, we designed obstacles as curved point cloud surfaces using the open-source software Blender[5] and used them to substitute parts of the road. To avoid modeling the obstacles by hand, we followed an automated procedure to generate a plethora of different obstacles based on several parameters, such as depth, ellipticity and size. An example of a frame in the CARLA simulator with a simulated pothole is presented in Fig. 4 (a) (texture) and in Fig. 4 (b) (geometry).

IV. Interfaces and Communication

Context-awareness is a critical factor for successful take-over requests and a lot of effort has been devoted to determining the type of stimulus (e.g. visual, auditory, vibrotactile) [59] and the required time-window [60], [61], [62]. In the case of partial or conditional driving automation, our

[5] https://www.blender.org/

framework could be used to prepare the driver to quickly take the control of the vehicle, if requested. In order to ensure that the driver is able to swiftly take over the control of the vehicle in an efficient way, we developed a notification system that presents relevant information about the condition of the environment. Our notification system is based on non-intrusive visual cues to prevent tunnel visioning, alerting the driver of potential risks and also directing his/her attention to the objects of interest that sparked the take-over request. In that way, in addition to assisting the human operator during manual driving, the system can, in times of automated driving, trigger the attention of the operator to possible external hazards and preparing him/her to resume control. The visualization technique presented in this section is designed as an AR windshield interface, although this is not restrictive, i.e. the method can be implemented in any AR interface.

A. AR Visualization

The visualization of obstacles is performed by projection. Assuming the position is known for the AR interface and the LiDAR relative to the world, we construct a transformation matrix to map the points of the point cloud from the LiDAR relative coordinate system to the AR interface's coordinate system. The transformation between two different coordinate systems is typically performed by applying serially a scale, a rotation and then a translation transformation. Since both coordinate systems are orthonormal, the scaling can be omitted. Also, by taking advantage of the rigid body nature of the vehicle where the LiDAR and AR interface is located, we also omit the rotation matrix given that, without loss of generality, we can assume that the two coordinate systems are aligned. According to these assumptions, the LiDAR coordinates are transformed into the AR interface's coordinates by a simple translation.

For projecting the points of the point cloud to the AR interface, we assume a simple pinhole camera model. If the AR interface is, for example, an AR windshield, then the windshield represents the image plane and the head of the driver the principal point with coordinates (x_0, y_0). That way, the focal distance $f = (f_x, f_y)$ represents the distance from the driver to the image plane. With the dimensions of the image plane (windshield), and specifically the aspect ratio, known, the frustum is fully defined and the projection can be made from a point in 3D windshield coordinates (x, y, z) to pixels (u, v) on the image plane using the following equation:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

An undesirable property is the sparsity of the projected pixels attributed to the sparsity of the point cloud. To overcome this limitation, we use an iterative nearest neighbour algorithm on the image space to fill the gaps between projected points. The result of this process is shown in Figs. 4 (d)-(i).

More specifically, Fig. 4 (d) and Fig. 4 (g) illustrate the segmented point cloud projected to the AR interface of ego1 and ego2 correspondingly. Note that all information is rendered for the sake of completeness. In real-world cases only the necessary information (e.g., arrows or recognised potholes) will be rendered so as to avoid clutter. Fig. 4 (e) shows the perspective projection of the points to the AR interface and image filling for ego1. In Fig. 4 (f) and Fig. 4 (i), the pothole recognition and visualization is depicted for the vehicles ego1 and ego2, while a warning about an upcoming pothole (retrieved from the database) before reaching the field of view of ego2 is presented in Fig. 4 (h).

We would like to clarify here that for evaluation of our methodology and demonstration purposes in the previous figures we project and illustrate in the 2D display device all the information from scene segmentation. However, in real driving scenarios only the most relevant information of the scene (e.g., dangerous objects, potholes) would be highlighted and displayed so as to decrease the amount of any unnecessary information that may bother or confuse the driver.

B. Information Storage and Vehicle Communication Rules

One of the advantages of autonomous vehicles is their ability to communicate with each other forming a cyber-physical system of systems. Many new opportunities arise from the ability of systems to share information, one of which is the transmission of objects or landmarks of interest that were previously observed by an agent, to other agents of the system who could benefit from such information. In particular, our work focuses on information sharing among vehicles about encountered obstacles, such as potholes and bumps, through a centralized server. When a vehicle identifies an unexpected (i.e., unregistered) obstacle, the vehicle sends a request to the server and after further inspection, the new potential obstacle is either discarded or added to the database. Vehicles may also send information regarding already known obstacles when they come across them. Such information includes the Global Positioning System (GPS) location, dimensions and geometrical characteristics in case the obstacle needs updating in the database, e.g. it has increased in size or has been fixed. Through this communication system, a driver can be warned about potential hazards that may not yet be in his field of view or they are obstructed by other objects and thus, increase his performance and decision-making abilities. We should clarify that our work does not focus on communication protocols and defence mechanisms against potential network attacks, but rather defines a solid framework describing the roles of each node and the information flow.

By using the LiDAR-based obstacle detection method, described in section III, the vehicle transmits via a communication component to a central server the points belonging to the obstacle, segmented from the point cloud scene. The information is coupled with a timestamp and the GPS location of the vehicle at that instance. The server then transmits to any vehicle in the vicinity of the obstacle, alerting (autonomous vehicles or human operators) about potential hazards from a large distance and thus helping alleviate the inability of the LiDAR sensor to identify obstacles from such a range. In the case of a driver, we also use the AR interface of the vehicle to display, in a non-distracting manner, the location and nature of the potentially upcoming obstacle.

Potholes can change shape over time, most commonly due to deterioration of the surrounding pavement and erosion caused by environmental effects or in the opposite case due to pothole repair. Thus, periodic updates are necessary for the long-term reliability of the pothole visualization component. As there is a need for periodical evaluation of the objects in the server database and update in the case of changes, we assign a shape- and geometry-based descriptor at each obstacle, so that it is characterized by a unique representative signature. Thus, every vehicle encountering the obstacle in a nearby range, calculates the descriptor of the obstacle's area. The new descriptor is then transmitted to the server and is used to confirm whether the information is up-to-date. In the case of a difference in the descriptor's value, an algorithm running in the server decides between keeping the old descriptor, updating it with the new one, or marking the obstacle as removed and deleting the entry from the database.

More specifically, we implement a simple system that (when a new pothole is detected) initiates a database search to retrieve whether the pothole is new or already existed and needs to be updated. Since potholes are static and thus change only in shape, the similarity check is based only on the bounding box of the re-identified pothole. When the overlap of the bounding boxes is less than a threshold, the previous object is replaced by the new one. In our experiments we used a threshold of 15% reshape in the area in either direction to avoid frequent unnecessary updates, while also retaining the required precision in representation. Similarly, the algorithm checks for significant changes in the bounding box dimensions. A flowchart showcasing the information update and communication pipeline between two vehicles is shown in Fig. 5.

V. Experimental Analysis

In this section, we will present and discuss in detail the experimental analysis and will evaluate our proposed framework.

A. Experimental Setup, Datasets and Metrics

The experiments were carried out on an Intel Core i7-4790HQ CPU @ 3.60GHz PC with 16 GB of RAM. The core algorithms are written in Matlab and C++. The evaluation of the methodology was performed using (i) synthetic dataset of potholes that we have created and (ii) 3D point cloud potholes from real datasets with known models (used as ground truth) which have been evaluated by other methods too [9], [63], [64], [65].

The pothole detection algorithms are compared in terms of the pixel-level (for image-based methods) and point-level (for point clouds) $precision = [TP/(TP+FP)]$, $recall = [TP/(TP+FN)]$, $accuracy = [(TP+TN)/(TP+TN+FP+FN)]$ and $F-score = 2 \cdot [(precision \cdot recall)/(precision+recall)]$, where TP, FP, TN, FN, represent the number of True-Positive, False-Positive, True-Negative and False-Negative pixels, respectively. The positive class includes all vertices belonging to the pothole (P) and the negative class all vertices belonging to the road (R). The performance metrics can also be expressed as shown in Table I, where *Real Pothole* (RP) represents the recall or in other words the percentage of vertices correctly annotated as pothole, *Real Road* (RR) represents the percentage of vertices correctly annotated as road, *Not real Pothole* (NP) represents the percentage of vertices wrongly annotated as pothole and *Not real Road* (NR) represents the percentage of vertices wrongly annotated as road.

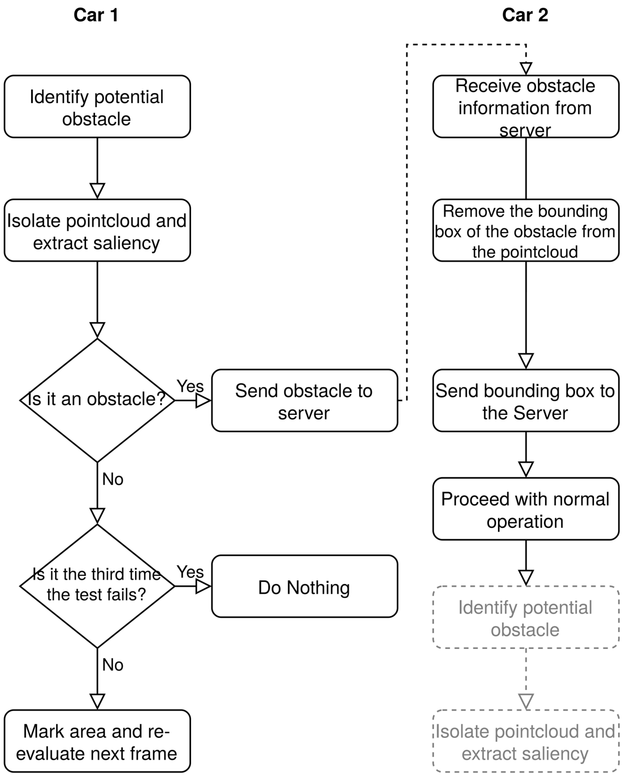

Car 1
Car 2
Identify potential obstacle
Isolate pointcloud and extract saliency
Is it an obstacle?
Yes
Send obstacle to server
No
Is it the third time the test fails?
Yes
Do Nothing
No
Mark area and re-evaluate next frame
Receive obstacle information from server
Remove the bounding box of the obstacle from the pointcloud
Send bounding box to the Server
Proceed with normal operation
Identify potential obstacle
Isolate pointcloud and extract saliency

Fig. 5. Flowchart of communicative vehicles for obstacle sharing.

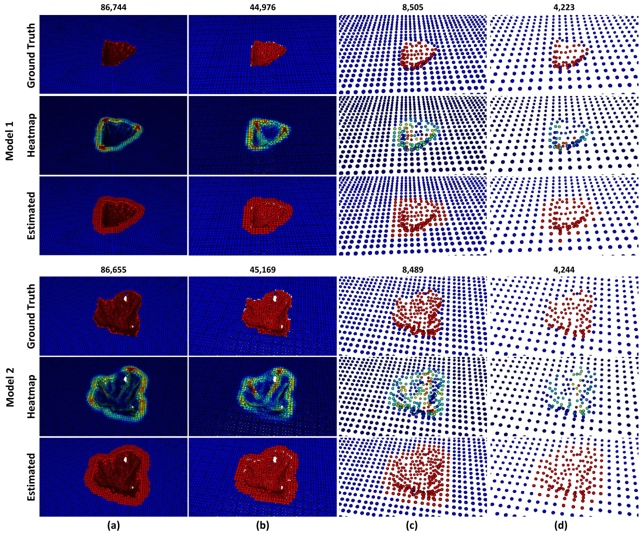


Fig. 6. Pothole detection in point cloud data of real potholes [63]. Two dense models are visualized: model1 (rows 1-3) and model2 (rows 4-6). For each model, the three rows illustrate (i) the ground truth, (ii) the heatmap visualizing the saliency map of the pothole and (iii) the estimated point cloud, respectively. The columns show results with decreasing density resolutions (in respect to the original model): (a) original model, (b) $\sim$ 50% of the vertices, (c) $\sim$ 10% of the vertices, (d) $\sim$ 5% of the vertices.

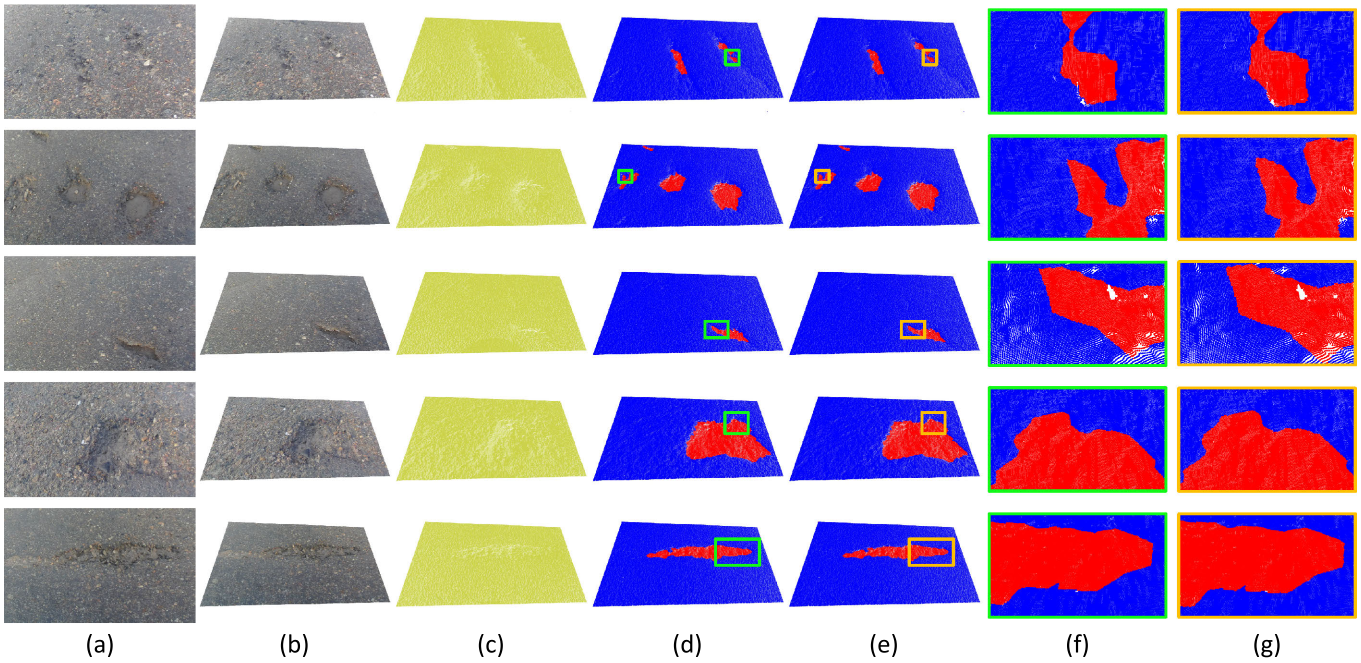


Fig. 7. Pothole detection on real data [9]. (a) RGB images of potholes, (b) corresponding point cloud of potholes with texture, (c) point cloud of potholes, (d) ground truth binary mask of potholes, (e) estimated binary mask of potholes, (f) enlarged details of the ground truth point cloud, (g) enlarged details of the estimated point cloud.

TABLE I

EVALUATION METRICS FOR POTHOLE DETECTION (IN PERCENTAGE %)

$(\times 100\%)$	Annotated as Pothole	Annotated as Road
Actual Pothole	$\mathrm{RP} = \frac{\mathrm{TP}}{\mathrm{P}}$	NR = 1 - RP
Actual Road	NP = 1 - RR	$\mathrm{RR} = \frac{\mathrm{TN}}{\mathrm{R}}$

B. Results

A quantitative comparison of our approach to other methods is challenged by the absence of datasets containing labeled and real potholes in point cloud format. Furthermore, the existing LiDAR-based methods [68], [69], [70] in the literature are still in their early stages, often using 2D LiDAR data, and they do not provide any available open datasets for quantitative comparison. Therefore, the only way to compare our pothole detection method was with image-based approaches that provide the corresponding point clouds for each pothole image. Despite the distinction in detection techniques between point clouds and RGB images, we maintained consistency by using the same models and metrics to ensure a fair comparison.

For the evaluation of our method, two public available datasets [9], [63] were utilized providing point clouds of real potholes. Fig. 6 visualizes results of our pothole detection method for the dataset created by real potholes [63] under different density resolutions (Fig. 6 (a)-(d)). Points in red represent the vertices belonging to the pothole, while points in blue represent vertices belonging to the road, both for the ground truth and the estimated point clouds. Two dense models (Fig. 6 (a)) are utilized as presented in rows 1-3 and 4-6, respectively. To investigate the performance of our approach in more realistic conditions, we increasingly downsampled the original point cloud (Fig. 6 (b)-(c)) to evaluate the robustness of detection of our algorithm. The corresponding number of vertices for the two models (original and downsampled) are shown above each model, respectively. The heatmap (rows 2 and 5) illustrates the geometric and spectral saliency per vertex (as estimated from Eq. 8). Higher salient values are depicted with deep red color while lower salient values with deep blue.

Due to the sensitive nature of the specific application involving safety of drivers (via information visualization for situational awareness), we prefer our algorithm to provide a small percentage of NR than having even a small value of NP (please refer to Table I). To wrongly identify as a pothole a small area of the road around an actual pothole is not as critical in our application as the opposite, namely to fail to present or partially present a potentially dangerous object (e.g., pothole, ramp). The detailed results with all evaluation metrics are shown in Table II for each of the thirteen 3D models of the point cloud dataset, and under different point cloud density resolutions. The results of this table show that our method is robust even for very low point cloud density. This is an important observation, since the output of the LiDAR device has a low density resolution pattern.

Fig. 7 visualizes some examples of the pothole detection algorithm applied in an other dataset [9]. The first column of this figure illustrates the RGB image presenting real road potholes. In the second column (Fig. 7-(b)), the corresponding point cloud with the relative texture is presented. The geometry represented by the 3D coordinates of the point cloud (without any color information) is presented in Fig. 7-(c). Fig. 7-(d) shows the ground truth vertices (in red) representing the pothole and Fig. 7-(e) presents our pothole estimation result. Figs. 7-(f) & (g) just present enlarged details of Figs. 7-(d) & (e), respectively, for easier visual comparison.

TABLE II

POTHOLE DETECTION ACCURACY (IN PERCENTAGE %) FOR DIFFERENT DENSITY RESOLUTIONS OF THE POINT CLOUD MODELS

Models	Original		$\sim 0.5 *$ Original		$\sim 0.1 *$ Original		$\sim 0.05 *$ Original	
Model 1	RP = 100	NR = 0	RP = 100	NR = 0	RP = 100	NR = 0	RP = 100	NR = 0
	NP = 0.44	RR = 99.56	NP = 0.40	RR = 99.60	NP = 0.68	RR = 99.32	NP = 0.89	RR = 99.11
Model 2	RP = 100	NR = 0	RP = 100	NR = 0	RP = 99.38	NR = 0.62	RP = 100	NR = 0
	NP = 0.65	RR = 99.35	NP = 0.58	RR = 99.42	NP = 0.98	RR = 99.02	NP = 1.22	RR = 98.78
Model 3	RP = 100	NR = 0	RP = 100	NR = 0	RP = 100	NR = 0	RP = 100	NR = 0
	NP = 0.41	RR = 99.59	NP = 0.36	RR = 99.64	NP = 0.63	RR = 99.37	NP = 0.74	RR = 99.26
Model 4	RP = 99.91	NR = 0.08	RP = 100	NR = 0	RP = 100	NR = 0	RP = 100	NR = 0
	NP = 0.45	RR = 99.55	NP = 0.41	RR = 99.59	NP = 0.76	RR = 99.24	NP = 0.93	RR = 99.07
Model 5	RP = 100	NR = 0	RP = 100	NR = 0	RP = 99.36	NR = 0.64	RP = 100	NR = 0
	NP = 0.62	RR = 99.38	NP = 0.56	RR = 99.44	NP = 0.86	RR = 99.14	NP = 1.25	RR = 98.75
Model 6	RP = 100	NR = 0	RP = 99.57	NR = 0.43	RP = 100	NR = 0	RP = 100	NR = 0
	NP = 0.52	RR = 99.48	NP = 0.46	RR = 99.54	NP = 0.78	RR = 99.22	NP = 1.22	RR = 98.78
Model 7	RP = 99.92	NR = 0.08	RP = 100	NR = 0	RP = 99.15	NR = 0.85	RP = 100	NR = 0
	NP = 0.47	RR = 99.53	NP = 0.44	RR = 99.56	NP = 0.80	RR = 99.20	NP = 1.10	RR = 98.90
Model 8	RP = 100	NR = 0	RP = 100	NR = 0	RP = 100	NR = 0	RP = 100	NR = 0
	NP = 0.48	RR = 99.52	NP = 0.44	RR = 99.56	NP = 0.73	RR = 99.27	NP = 1.06	RR = 98.94
Model 9	RP = 100	NR = 0	RP = 99.13	NR = 0.87	RP = 100	NR = 0	RP = 100	NR = 0
	NP = 0.45	RR = 99.55	NP = 0.37	RR = 99.63	NP = 0.65	RR = 99.35	NP = 0.96	RR = 99.04
Model 10	RP = 99.92	NR = 0.08	RP = 100	NR = 0	RP = 100	NR = 0	RP = 100	NR = 0
	NP = 0.42	RR = 99.58	NP = 0.38	RR = 99.62	NP = 0.68	RR = 99.32	NP = 0.84	RR = 99.16
Model 11	RP = 100	NR = 0	RP = 100	NR = 0	RP = 100	NR = 0	RP = 98.56	NR = 1.44
	NP = 0.59	RR =99.41	NP = 0.51	RR = 99.49	NP = 0.95	RR = 99.05	NP = 1.03	RR = 98.97
Model 12	RP = 100	NR = 0	RP = 99.36	NR = 0.64	RP = 100	NR = 0	RP = 100	NR = 0
	NP = 0.44	RR = 99.56	NP =0.36	RR = 99.64	NP = 0.64	RR = 99.36	NP = 0.81	RR = 99.19
Model 13	RP = 99.96	NR = 0.04	RP = 100	NR = 0	RP = 100	NR = 0	RP = 100	NR = 0
	NP = 0.42	RR = 99.58	NP = 0.38	RR = 99.62	NP = 0.65	RR = 99.35	NP = 0.79	RR = 99.21
Average	RP = 99.98	NR = 0.02	RP = 99.85	NR = 0.15	RP = 99.83	NR = 0.17	RP = 99.88	NR = 0.11
	NP = 0.49	RR = 99.51	NP = 0.43	RR = 99.57	NP = 0.75	RR = 99.25	NP = 0.99	RR = 99.01%

TABLE III

COMPARISON OF THE POTHOLE DETECTION ACCURACY AMONG DIFFERENT STATE-OF-THE-ART APPROACHES

Dataset	Method	Correct Detection	Incorrect	Misdetection	Recall	Precision	Accuracy	F-score
1	1 [68]	11	11	0	0.520	0.543	0.989	0.531
	2 [69]	22	0	0	0.462	0.998	0.994	0.632
	3 [65]	22	0	0	0.499	0.987	0.994	0.663
	4 [64]	21	1	0	0.701	0.964	0.995	0.811
	our	22	0	0	0.853	0.993	0.991	0.918
2	1 [68]	42	10	0	0.975	0.971	0.999	0.973
	2 [69]	40	8	4	0.874	0.991	0.997	0.929
	3 [65]	51	1	0	0.980	0.980	0.999	0.980
	4 [64]	52	0	0	0.950	0.883	0.992	0.915
	our	40[2]	0	0	0.909	0.996	0.992	0.951
3	1 [68]	5	0	0	0.612	0.771	0.995	0.683
	2 [69]	5	0	0	0.534	0.992	0.996	0.694
	3 [65]	5	0	0	0.582	0.983	0.996	0.731
	4 [64]	5	0	0	0.702	0.996	0.996	0.823
	our	5	0	0	0.953	0.984	0.996	0.969
Total	1 [68]	58	21	0	0.800	0.822	0.994	0.800
	2 [69]	67	8	4	0.695	0.992	0.995	0.817
	3 [65]	78	1	0	0.771	0.982	0.996	0.864
	4 [64]	78	1	0	0.890	0.898	0.996	0.894
	our	67	0	0	0.899	0.994	0.992	0.945

[2] Only 40 of the refereed (52) models were founded online.

Table III provides a qualitatively comparison of our method versus other approaches of the literature. However, it should be mentioned that the results are not directly comparable because the other methods use only the visual information of the RGB images, while our method uses only the geometrical information of the corresponding point cloud.

The experimental analysis showed that our methodology is superior to image-based pothole detection methodologies. This may be attributed to the use of LiDAR sensing, which provides depth information and has a large range, making it suitable for driving environments. In contrast, image-based methods rely on visual cues, such as changes in color or texture, which can be ambiguous and difficult to interpret. Moreover, the depth of a pothole cannot be accurately estimated by 2D information alone. LiDAR sensors can provide 3D information about the road surface, enabling more accurate detection and localization of potholes. This allows accurate identification and localization of potholes even when the potholes are partially obscured by other objects or difficult to identify in images. Additionally, LiDAR sensors are less affected by changes in lighting conditions, such as shadows and reflections, which can impact the performance of image-based methods. They also have a longer range than cameras, enabling obstacles' detection at greater distances, and providing in this way drivers with more advanced warning of potential hazards, thereby improving overall safety. The point cloud processing system analyzes the environmental data and classifies regions as safe or potentially hazardous based on the presence of obstacles. A flexible visualization system was implemented to prioritize the display of critical information in the driver's field of view and accordingly adjust the properties of rendered information. This ensures that the driver is alerted to potential hazards in a non-intrusive way, without being overloaded with unnecessary information.

C. Visualization and AR Rendering

The detected potholes are rendered on an AR display in the driver's field of view, which can be a head-up display (HUD) or a windshield display. The rendering of potholes can be overlaid on the real-world view, using a variety of visual cues to indicate their severity. For example, potholes that are more salient could be rendered in larger sizes, brighter colours, or with animated effects, while less dangerous potholes could be rendered in smaller sizes or subtler colours.

To ensure the safe and effective conveying of information about potholes, the AR visualization system should be designed to be non-intrusive and appropriate for the driving context. We have implemented a flexible visualization system (Figs. 8-9) that allows to select a number of properties for the obstacles to be rendered (size, colour, animation and visualization type), such as to maximize the driver's ability to realize the presence of hazardous obstacles while minimizing distraction. Next, we first present some general recommendations on the choice of optimal properties for the visualization system, and upon this we present our preliminary research on the evaluation of users' personal preferences for customization of visualization properties in a simulated driving environment.

To present information in a non-intrusive way, it is important to strike a balance between providing enough information to enhance situational awareness while avoiding information overload that may distract the driver. Some strategies for presenting AR information in a non-intrusive manner are: (a) Prioritize information [71] that is critical for safe driving, such as upcoming obstacles or hazards. (b) The size and placement of information can affect how noticeable it is to the driver [72]. Important information should be presented in a larger size and placed in a location where it can be easily seen without distracting the driver from the road. (c) Colours can affect how noticeable and attention-grabbing information is [72]. High-contrast colours, such as red, can be effective for highlighting critical information, while more muted colours can be used for less important information. (d) Animation can be attention-grabbing, but it can also be distracting. Animated information should be used sparingly and only for critical information that requires immediate attention [73]. (e) Drivers have different preferences for how they want information presented to them [74]. Providing customization options for the size, colour, and placement of information can help drivers tailor the information to their preferences and reduce distraction. Feedback from drivers can help identify areas where the presentation of information can be improved and help strike the right balance between providing enough information and not distracting the driver.

- **Size**: The size of the rendered obstacles should be proportional to their saliency. More important obstacles, such as potholes, should be rendered larger than less important ones. However, it is important to avoid rendering the obstacles so large that they become intrusive or distracting. The size of the rendered obstacles should be designed to quickly grab the driver's attention and provide sufficient information for the driver to take appropriate action.
- **Color**: The use of color can also help convey the saliency of obstacles. Brighter colors, such as red or yellow, can be used to indicate high saliency, while less important obstacles can be rendered with subtler colors. The use of contrasting colors can also help the driver quickly differentiate between obstacles and the surrounding environment.
- **Animation**: Animation can be used to provide additional visual cues that indicate the saliency of obstacles. For example, obstacles that are in the driver's immediate path could be rendered with a flashing or pulsating effect, while less important obstacles could be rendered with subtler animation.
- **Visualization type**: The AR content may be provided through different visualization techniques (Fig. 8):
  - Wedge 3D. This method is based on rendering objects in a form of a pyramid-like scheme. The height of the pyramid is proportional to the distance between the user's vehicle reference and the object.
  - Arrow. This visualization method involves a stick and an arrow tip. The arrow's direction (in 3D) follows the object's position, while the length of the stick is proportional to the corresponding distance.
  - 3D Minimap. Three layers of concentric spheres are used to provide an estimation of the object's distance from the vehicle's position.
  - Radar. The objects are represented as small squares in a radar-like area.

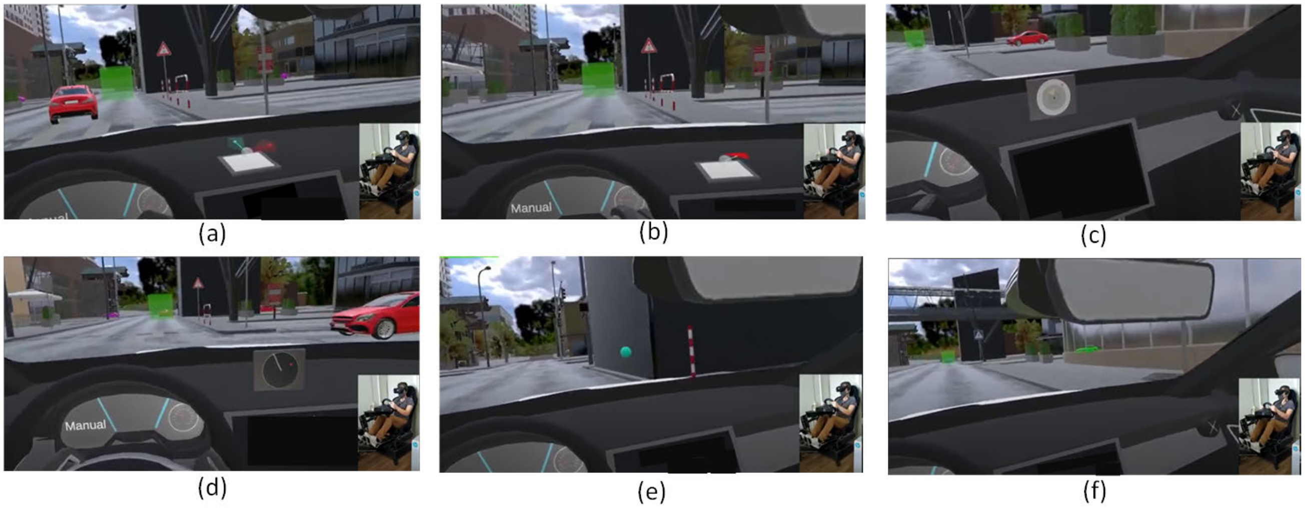


Fig. 8. Visualization using (a) 3D wedge, (b) 3D arrows, (c) 3D minimaps, (d) Radar, (e) 3D sphere representing the object of interest, (f) 3D mesh overlay.

- Meshes Sphere. Similar to the previous method, however, the occluded objects are represented by spheres.
- Occluded Meshes As is. The occluded information is presented at the appropriate distance by transparently rendering the object's silhouette.

Additionally to the visual symbolic warnings, other alternative modalities can be used as well. For example when a vehicle is approaching a pothole that has been detected by another vehicle, it could receive a text or sound warning to slow down or take immediate action.

1) Rule-Based Visualization and Warnings: In terms of visualizing and representing the outcomes of the proposed method through AR rendering, we suggest using a colour-coded system to indicate the severity of the detected potholes. For example, green could represent a small or shallow pothole, red could indicate a larger or deeper pothole that poses a greater risk to the driver, while orange could represent an intermediate situation (Fig. 9). The size and shape of the rendered pothole could also be adjusted to reflect its severity and distance from the driver. To ensure that the information is presented in a non-intrusive way, we suggest limiting the amount of information displayed on the AR-HUD to only the most critical or relevant obstacles. This can be achieved through intelligent filtering and prioritization algorithms that take into account factors such as the driver's speed and direction of travel, as well as the severity and proximity of detected obstacles (like the Pavement Condition Index (PCI) [75] and the International Roughness Index (IRI) [76]). By combining the IRI with the depth and size of potholes, a more comprehensive measure of the severity of the road surface condition can be obtained. By providing only the most relevant information, we can help reduce the risk of information overload and ensure that the driver's attention remains focused on the road ahead.

2) User Evaluation Study: In order to design the visualization scheme following user-centered principles, a user evaluation study was performed to assess personal preferences while driving in a simulated environment. A steering wheelchair was implemented for the driving simulation, and a VR display device with leap motion sensor was used to provide the AR information (Fig. 8). Out of all properties, we focused on the customization of the visualization type which is considered more critical and less subjective than other factors like size, color, and animation.

A total of 12 adult (4 females, 8 males) participants took part in the experiment. Most of them were either employees or students at the University of Patras. The age range of the participants was 23 to 45 years with an average value of 27.5 years. The experimental process consisted of two parts. In the first part, the participants familiarized themselves with the simulator and the driving process. They were not given any specific instructions or time constraints, and were free to drive in any direction and for any duration they felt comfortable with. During this phase, the simulated vehicles adhered to driving rules and moved safely on the road. The participants had the opportunity to explore the simulator environment using all the provided visualization methods. In the second part, the participants were asked to follow a predetermined route that included various hazardous situations based on predefined scenarios. The different visualization methods available were utilized to enhance the drivers' situational awareness. Spatial indicators were used to guide them along the correct path. Finally, a questionnaire was administered at the end of the experimental process, which included general demographic questions (gender, age, education, etc.), inquiries about technology usage frequency, and questions pertaining to the evaluation of the visualization techniques. Specifically, the participants evaluated whether the information provided through the visualization techniques met their personal expectations, increased their trust and acceptance, and whether it was understandable and non-distracting.

The most important outcomes of the evaluation study are summarized next. All participants agreed that utilizing the proposed visualization system would enhance their driving safety and awareness of critical upcoming events. They also found the tool particularly useful when navigating unfamiliar

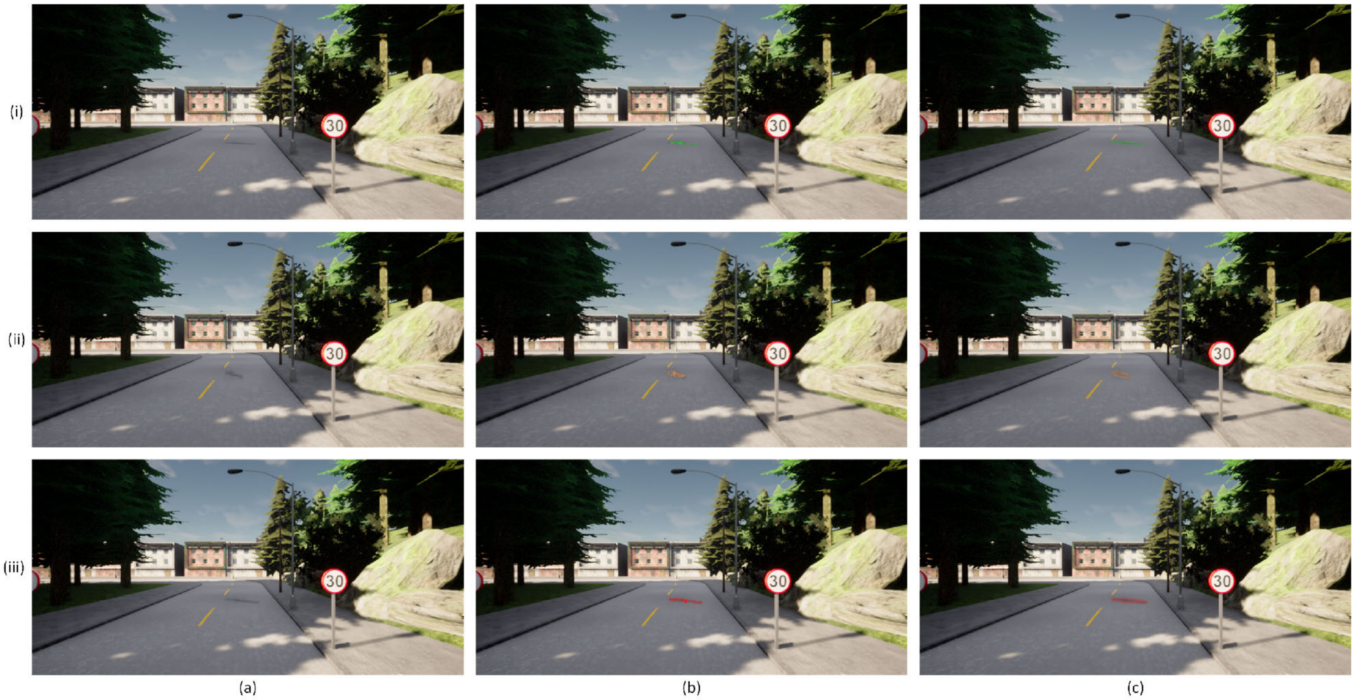


Fig. 9. (i-iii) Different severity prioritization of potholes based on their size and volume, (a) image of the camera without any visual cue, (b) projected point cloud representing the identified pothole, (c) overlay visualization of the pothole.

mixed traffic environments. Furthermore, a significant majority of participants (83.3%) expressed a positive inclination towards using the proposed visualization system to alleviate their nervousness while driving in unknown areas. Regarding the application's efficacy in promoting safe and secure driving, all participants were in favor (66.7% strongly agree, 33.3% somewhat agree) of utilizing it. Nearly all participants (11 out of 12) demonstrated awareness of VR/AR training tools, and all of them (100%) expressed interest in utilizing VR/AR technology for training or learning purposes. It should be however mentioned, that the characteristics of the group of participants might not reflect that of the average population, since the majority of participants (75%) had prior experience using an AR device, tool, or application, and all of them (100%) have previously used a VR device, tool, or application.

With regards to visualization type, attendees had diverse preferences. The most popular method was the presentation of occluded objects as transparent meshes, and the display of the object's silhouette in their true form (83.3% rated it from 4 to 5 on a scale of 1 to 5). On the other hand, the least popular method was the use of minimaps (50% rated it below 3 on a scale of 1 to 5).

VI. Conclusion

In this work, we propose a cooperative obstacle detection and rendering scheme that utilizes LiDAR data and driving patterns to identify obstacles within the road range. Our system allows for information sharing between connected vehicles, enabling drivers to be notified about incoming potholes even when there is no direct line-of-sight. This cooperative driving scheme increases situational awareness and reduces the risk of accidents caused by unexpected obstacles. Our method is based on the analysis of point clouds which is challenged by the lack of benchmark datasets obtained from LiDAR devices. To overcome this problem, we created our own synthetic dataset and added it to the maps of the CARLA simulator, thereby creating realistic driving environments. The comparison of our method with other state-of-the-art approaches, regarding the accuracy of pothole detection in real datasets, has shown its effectiveness providing very promising outcomes. Our proposed approach can be extended to cover a wider range of road hazards beyond potholes, such as debris or uneven road surfaces. By utilizing the same LiDAR sensor technology, we can detect these hazards and provide similar AR visualizations to drivers. Moreover, we plan to investigate the integration of other sensing modalities, such as RGB-D cameras, which could provide additional visual information to improve the accuracy of obstacle detection and enhance the situational awareness of drivers. In addition, our methodology can be further improved by incorporating machine learning algorithms to enhance the accuracy and efficiency of obstacle detection and classification. We plan to explore the use of deep learning models, which have shown promising results in various computer vision tasks, to enhance our point cloud processing system. Lastly, we envision that our proposed approach could be applied beyond personal vehicles, such as in autonomous vehicles and public transportation systems. By leveraging V2X communication, our cooperative obstacle detection and rendering scheme could provide a safer driving experience for all road users.

Appendix A
Robust Principal Component Analysis (RPCA)

RPCA is a powerful mathematical tool that has been used in many scientific domains in order to decompose an observed measurement $\mathbf{E}$ into a low-rank matrix $\mathbf{L}$, representing the ideal data unaffected by any kind of noise, and a sparse matrix $\mathbf{S}$, representing the noisy data. Decomposition is performed by solving the following equation:

$$\underset{\mathbf{L},\mathbf{S}}{\operatorname{argmin}} \ \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1, \quad \text{s.t. } \mathbf{L}+\mathbf{S}=\mathbf{E}, \tag{9}$$

where $\|\mathbf{L}\|_*$ is the nuclear norm of a matrix $\mathbf{L}$ (i.e, $\sum_i \sigma_i(\mathbf{L})$ is the sum of the singular values of $\mathbf{L}$).

A lot of works have been proposed all of these years, presenting excellent results. However, despite the effectiveness that some works [77], [78] have presented in the past, the execution times of the proposed algorithms need improvement. This convex problem can be solved using a very fast approach, as described in [79], according to:

$$\underset{\mathbf{L},\mathbf{S}}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{L}+\mathbf{S}-\mathbf{E}\|_F + \lambda\|\mathbf{S}\|_1 \quad \text{s.t. } \operatorname{rank}(\mathbf{L}) = K \tag{10}$$

$$\mathbf{L}^{(t+1)} = \underset{\mathbf{L}}{\operatorname{argmin}} \ \|\mathbf{L}+\mathbf{S}^{(t)}-\mathbf{E}\|_F \quad \text{s.t. } \operatorname{rank}(\mathbf{L}) = K \tag{11}$$

$$\mathbf{S}^{(t+1)} = \underset{\mathbf{S}}{\operatorname{argmin}} \ \|\mathbf{L}^{(t+1)}+\mathbf{S}-\mathbf{E}\|_F + \lambda\|\mathbf{S}\|_1 \tag{12}$$

In each (t) iteration, the Eq. (11) is updated with rank $= K$. If $\frac{u_K}{\sum_{i=1}^{K} u_i} > \epsilon$, where u denotes the singular values and ϵ is a small threshold, then the rank is increased by one (i.e., $K = K+1$) and the Eq. (12) is updated too. To update the Eq. (11), a partial SVD$(\mathbf{E}-\mathbf{S}^{(t)})$ is estimated keeping K components. To update the Eq. (12), a shrinkage operator is used $\mathcal{D}(.)$, where:

$$\mathcal{D}(\mathbf{E}-\mathbf{L}^{(t+1)}, \lambda) = \operatorname{sign}(\mathbf{E}-\mathbf{L}^{(t+1)}) \max\{0, |\mathbf{E}-\mathbf{L}^{(t+1)}| - \lambda\} \tag{13}$$

References

[1] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for on-Road Motor Vehicles*, SAE Standard J3 016 201 806, 2018.

[2] J. Leng, Y. Liu, D. Du, T. Zhang, and P. Quan, "Robust obstacle detection and recognition for driver assistance systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1560–1571, Apr. 2020.

[3] T. L. Overton, T. E. Rives, C. Hecht, S. Shafi, and R. R. Gandhi, "Distracted driving: Prevalence, problems, and prevention," *Int. J. Injury Control Saf. Promotion*, vol. 22, no. 3, pp. 187–192, Jul. 2015, doi: 10.1080/17457300.2013.879482.

[4] S. McEvoy and M. Stevenson, "An exploration of the role of driver distraction in serious road crashes," in *Proc. Int. Conf. Distractions Driving*, Sydney, NSW, Australia, 2007, pp. 189–211.

[5] I. Dua, A. U. Nambi, C. V. Jawahar, and V. Padmanabhan, "AutoRate: How attentive is the driver?" in *Proc. 14th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2019, pp. 1–8.

[6] I. Radu and B. Schneider, "How augmented reality (AR) can help and hinder collaborative learning: A study of AR in electromagnetism education," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 9, pp. 3734–3745, Sep. 2023.

[7] A. Dhiman and R. Klette, "Pothole detection using computer vision and learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3536–3550, Aug. 2020.

[8] T. Sun, W. Pan, Y. Wang, and Y. Liu, "Region of interest constrained negative obstacle detection and tracking with a stereo camera," *IEEE Sensors J.*, vol. 22, no. 4, pp. 3616–3625, Feb. 2022.

[9] R. Fan, U. Ozgunalp, B. Hosking, M. Liu, and I. Pitas, "Pothole detection based on disparity transformation and road surface modeling," *IEEE Trans. Image Process.*, vol. 29, pp. 897–908, 2020.

[10] Y. He and Z. Liu, "A feature fusion method to improve the driving obstacle detection under foggy weather," *IEEE Trans. Transport. Electrific.*, vol. 7, no. 4, pp. 2505–2515, Dec. 2021.

[11] J. Dib, K. Sirlantzis, and G. Howells, "A review on negative road anomaly detection methods," *IEEE Access*, vol. 8, pp. 57298–57316, 2020.

[12] P. Sun, X. Zhao, Z. Xu, R. Wang, and H. Min, "A 3D LiDAR data-based dedicated road boundary detection algorithm for autonomous vehicles," *IEEE Access*, vol. 7, pp. 29623–29638, 2019.

[13] G. Wang, J. Wu, R. He, and B. Tian, "Speed and accuracy tradeoff for LiDAR data based road boundary detection," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 6, pp. 1210–1220, Jun. 2021.

[14] G. Wang, J. Wu, R. He, and S. Yang, "A point cloud-based robust road curb detection and tracking method," *IEEE Access*, vol. 7, pp. 24611–24625, 2019.

[15] L. Chen, J. Yang, and H. Kong, "LiDAR-histogram for fast road and obstacle detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 1343–1348.

[16] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp. 1–16.

[17] M.-K. Choi, J.-H. Lee, H. Jung, I. R. Tayibnapis, and S. Kown, "Simulation framework for improved UI/UX of AR-HUD display," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jul. 2018, pp. 1–4.

[18] N. Karatas et al., "Evaluation of AR-HUD interface during an automated intervention in manual driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 2158–2164.

[19] I. Schiopu, J. P. Saarinen, L. Kettunen, and I. Tabus, "Pothole detection and tracking in car video sequence," in *Proc. 39th Int. Conf. Telecommun. Signal Process. (TSP)*, 2016, pp. 701–706.

[20] A. Akagic, E. Buza, and S. Omanovic, "Pothole detection: An efficient vision based method using RGB color space image segmentation," in *Proc. 40th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2017, pp. 1104–1109.

[21] M. H. Asad, S. Khaliq, M. H. Yousaf, M. O. Ullah, and A. Ahmad, "Pothole detection using deep learning: A real-time and AI-on-the-edge perspective," *Adv. Civil Eng.*, vol. 2022, Apr. 2022, Art. no. 9221211, doi: 10.1155/2022/9221211.

[22] M. P. Jenkins and D. Young, "BARRACUDA: An augmented reality display for increased motorcyclist En route hazard awareness," in *Proc. IEEE Int. Multi-Disciplinary Conf. Cognit. Methods Situation Awareness Decis. Support (CogSIMA)*, Mar. 2016, pp. 68–72.

[23] D.-H. Heo, J.-Y. Choi, S.-B. Kim, T.-O. Tak, and S.-P. Zhang, "Image-based pothole detection using multi-scale feature network and risk assessment," *Electronics*, vol. 12, no. 4, p. 826, Feb. 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/4/826

[24] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection and classification using deep neural networks with smartphone images," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 12, pp. 1127–1141, Dec. 2018.

[25] Q. Mei, M. Gül, and M. R. Azim, "Densely connected deep neural network considering connectivity of pixels for automatic crack detection," *Autom. Construct.*, vol. 110, Feb. 2020, Art. no. 103018.

[26] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, "Feature pyramid and hierarchical boosting network for pavement crack detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1525–1535, Apr. 2020.

[27] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3434–3445, Dec. 2016.

[28] S. Gu, Y. Zhang, J. Yang, and H. Kong, "LiDAR-based urban road detection by histograms of normalized inverse depths and line scanning," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, 2017, pp. 1–6.

[29] G. Bosurgi, M. Modica, O. Pellegrino, and G. Sollazzo, "An automatic pothole detection algorithm using pavement 3D data," *Int. J. Pavement Eng.*, pp. 1–15, Apr. 2022.

[30] F. M. Ortiz, M. Sammarco, L. Henrique, M. K. Costa, and M. Detyniecki, "Vehicle telematics via exteroceptive sensors: A survey," 2020, *arXiv:2008.12632*.

[31] G. Arvanitis, A. S. Lalos, and K. Moustakas, "Saliency mapping for processing 3D meshes in industrial modeling applications," in *Proc. IEEE 17th Int. Conf. Ind. Informat. (INDIN)*, vol. 1, Jul. 2019, pp. 683–686.

[32] G. Arvanitis, A. S. Lalos, and K. Moustakas, "Robust and fast 3-D saliency mapping for industrial modeling applications," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1307–1317, Feb. 2021.

[33] L. Ma and J. Li, "SD-GCN: Saliency-based dilated graph convolution network for pavement crack extraction from 3D point clouds," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 111, Jul. 2022, Art. no. 102836.

[34] H. Wang et al., "Road boundaries detection based on local normal saliency from mobile laser scanning data," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 10, pp. 2085–2089, Oct. 2015.

[35] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2VNet: Vehicle-to-vehicle communication for joint perception and prediction," in *Computer Vision—ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, 2020, pp. 605–621.

[36] E. Arnold, M. Dianati, R. de Temple, and S. Fallah, "Cooperative perception for 3D object detection in driving scenarios using infrastructure sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 1852–1864, Mar. 2022.

[37] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, "OPV2V: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, May 2021, pp. 2583–2589.

[38] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3D point clouds," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*. Los Alamitos, CA, USA: IEEE Computer Society, Jul. 2019, pp. 514–524, doi: 10.1109/ICDCS.2019.00058.

[39] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira, "When2com: Multi-agent perception via communication graph grouping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 4105–4114.

[40] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds," in *Proc. 4th ACM/IEEE Symp. Edge Comput.* New York, NY, USA: Association for Computing Machinery, 2019, pp. 88–100, doi: 10.1145/3318216.3363300.

[41] J. Guo et al., "CoFF: Cooperative spatial feature fusion for 3-D object detection on autonomous vehicles," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11078–11087, Jul. 2021.

[42] Y. Yuan, H. Cheng, and M. Sester, "Keypoints-based deep feature fusion for cooperative vehicle detection of autonomous driving," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3054–3061, Apr. 2022.

[43] S. Fang, H. Li, and M. Yang, "LiDAR SLAM based multivehicle cooperative localization using iterated split CIF," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21137–21147, Nov. 2022.

[44] S.-W. Kim and W. Liu, "Cooperative autonomous driving: A mirror neuron inspired intention awareness and cooperative perception approach," *IEEE Intell. Transp. Syst. Mag.*, vol. 8, no. 3, pp. 23–32, Fall. 2016.

[45] W. Song, Y. Yang, M. Fu, F. Qiu, and M. Wang, "Real-time obstacles detection and status classification for collision warning in a vehicle active safety system," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 758–773, Mar. 2018.

[46] U. Ju, L. L. Chuang, and C. Wallraven, "Acoustic cues increase situational awareness in accident situations: A VR car-driving study," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3281–3291, Apr. 2022.

[47] J. A. Abbasi, D. Mullins, N. Ringelstein, P. Reilhac, E. Jones, and M. Glavin, "An analysis of driver gaze behaviour at roundabouts," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8715–8724, Jul. 2022.

[48] Q. Rao, C. Grünler, M. Hammori, and S. Chakraborty, "Design methods for augmented reality in-vehicle infotainment systems," in *Proc. 51st ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–6.

[49] D. L. Strayer, J. M. Cooper, R. M. Goethe, M. M. McCarty, D. J. Getty, and F. Biondi, "Assessing the visual and cognitive demands of in-vehicle information systems," *Cognit. Res., Princ. Implications*, vol. 4, no. 1, Dec. 2019.

[50] H. Kim and J. L. Gabbard, "Assessing distraction potential of augmented reality head-up displays for vehicle drivers," *Human Factors, J. Human Factors Ergonom. Soc.*, vol. 64, no. 5, pp. 852–865, Aug. 2022.

[51] L. Kettle and Y.-C. Lee, "Augmented reality for vehicle-driver communication: A systematic review," *Safety*, vol. 8, no. 4, p. 84, 2022. [Online]. Available: https://www.mdpi.com/2313-576X/8/4/84

[52] M. Dikmen and C. Burns, "Trust in autonomous vehicles: The case of Tesla autopilot and summon," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 1093–1098.

[53] D. Niu, J. Terken, and B. Eggen, "Anthropomorphizing information to enhance trust in autonomous vehicles," *Human Factors Ergonom. Manuf. Service Industries*, vol. 28, no. 6, pp. 352–359, Nov. 2018.

[54] L. Morra, F. Lamberti, F. G. Prattícó, S. L. Rosa, and P. Montuschi, "Building trust in autonomous vehicles: Role of virtual reality driving simulators in HMI design," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 9438–9450, Oct. 2019.

[55] P. Lindemann, T.-Y. Lee, and G. Rigoll, "Supporting driver situation awareness for autonomous urban driving with an augmented-reality windshield display," in *Proc. IEEE Int. Symp. Mixed Augmented Reality Adjunct (ISMAR-Adjunct)*, Oct. 2018, pp. 358–363.

[56] A. O. Yontem, K. Li, D. Chu, V. Meijering, and L. Skrypchuk, "Prospective immersive human-machine interface for future vehicles: Multiple zones turn the full windscreen into a head-up display," *IEEE Veh. Technol. Mag.*, vol. 16, no. 1, pp. 83–92, Mar. 2021.

[57] H. Kim, J. L. Gabbard, A. M. Anon, and T. Misu, "Driver behavior and performance with augmented reality pedestrian collision warning: An outdoor user study," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 4, pp. 1515–1524, Apr. 2018.

[58] (2023). *Asam Opendrive*. [Online]. Available: https://www.asam.net/standards/detail/opendrive/

[59] P. Bazilinskyy, A. Eriksson, S. Petermeijer, and J. de Winter, "Usefulness and satisfaction of take-over requests for highly automated driving," in *Proc. Road Safety Simul. Int. Conf.*, Oct. 2017, pp. 1–7.

[60] H. White, D. Large, D. Salanitri, G. Burnett, A. Lawson, and E. Box, "Rebuilding drivers' situation awareness during take-over requests in level 3 automated cars," *Proc. Conf., Ergonom. Hum. Factors, Stratford-Upon-Avon*, U.K., Apr. 2019.

[61] K. Zeeb, A. Buchner, and M. Schrauf, "What determines the take-over time? An integrated model approach of driver take-over after automated driving," *Accident Anal. Prevention*, vol. 78, pp. 212–221, May 2015.

[62] V. Melcher, S. Rauh, F. Diederichs, H. Widlroither, and W. Bauer, "Take-over requests for automated driving," *Procedia Manuf.*, vol. 3, pp. 2867–2873, Jul. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978915007891

[63] R. Fan, U. Ozgunalp, Y. Wang, M. Liu, and I. Pitas, "Rethinking road surface 3-D reconstruction and pothole detection: From perspective transformation to disparity map segmentation," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 5799–5808, Jul. 2022.

[64] R. Fan and M. Liu, "Road damage detection based on unsupervised disparity map segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4906–4911, Nov. 2020.

[65] R. Fan, X. Ai, and N. Dahnoun, "Road surface 3D reconstruction based on dense subpixel disparity map estimation," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3025–3035, Jun. 2018.

[66] Z. Zhang, X. Ai, C. K. Chan, and N. Dahnoun, "An efficient algorithm for pothole detection using stereo vision," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 564–568.

[67] U. Ozgunalp, X. Ai, and N. Dahnoun, "Multiple lane detection algorithm based on novel dense vanishing point estimation," *Signal, Image Video Process.*, vol. 10, no. 6, pp. 1127–1134, Sep. 2016.

[68] V. Prasad and S. Kumari, "Pothole detection using LiDAR," *Adv. Automobile Eng.*, vol. 10, no. 4, pp. 1–3, 2021.

[69] B.-H. Kang and S.-I. Choi, "Pothole detection system using 2D LiDAR and camera," in *Proc. 9th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2017, pp. 744–746.

[70] N. J. Sucgang, M. C. Ramos, and N. A. A. Arriola, "Road surface obstacle detection using vision and LiDAR for autonomous vehicle," in *Proc. Int. MultiConference Eng. Comput. Scientists*, 2017, pp. 1–5.

[71] M. Regan and R. Pose, "Priority rendering with a virtual reality address recalculation pipeline," in *Proc. 21st Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*. New York, NY, USA: Association for Computing Machinery, 1994, pp. 155–162, doi: 10.1145/192161.192192.

[72] J. Woodward and J. Ruiz, "Analytic review of using augmented reality for situational awareness," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 4, pp. 2166–2183, Apr. 2023.

[73] S. Sharma, J. Stigall, and S. T. Bodempudi, "Situational awareness-based augmented reality instructional (ARI) module for building evacuation," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces Abstr. Workshops (VRW)*, Jul. 2020, pp. 70–78.

[74] A. Riegler, A. Riener, and C. Holzmann, "Towards personalized 3D augmented reality windshield displays in the context of automated driving," *Frontiers Future Transp.*, vol. 3, Feb. 2022, Art. no. 810698. [Online]. Available: https://www.frontiersin.org/articles/10.3389/ffutr.2022.810698

[75] A. Issa, H. Samaneh, and M. Ghanim, "Predicting pavement condition index using artificial neural networks approach," *Ain Shams Eng. J.*, vol. 13, no. 1, Jan. 2022, Art. no. 101490. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2090447921002264

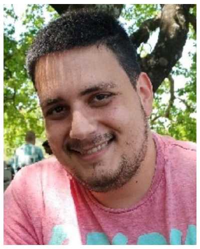
[76] P. R. Pawar, A. T. Mathew, and M. R. Saraf, “IRI (international roughness index): An indicator of vehicle response,” *Mater. Today, Proc.*, vol. 5, no. 5, pp. 11738–11750, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214785318303468

[77] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?” *J. ACM*, vol. 58, no. 3, p. 11, May 2011.

[78] Z. Lin, M. Chen, and Y. Ma, “The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices,” 2009, *arXiv:1009.5055*.

[79] P. Rodríguez and B. Wohlberg, “Fast principal component pursuit via alternating minimization,” in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 69–73.

Gerasimos Arvanitis received the Diploma degree from the Electrical and Computer Engineering Department, University of Patras, Greece, in 2009, and the M.Sc. degree in electronics and information processing and the Ph.D. degree in geometrical 3D processing from the University of Patras in 2011 and 2021, respectively. From 2011 to 2015, he was a Software Programmer with the Computer Technology Institute, Patras, Greece. He has participated in six European projects and he acts as a regular reviewer for several technical journals and conferences. He has (co)authored more than 50 papers in refereed journals, edited books, and international conferences. His current research interests include geometry processing of 3D models, reconstruction, compression, outliers removal, feature preserving denoising algorithms, and computer graphics. He has been a member of the Visualization and Virtual Reality Group, University of Patras, since January 2016.



Nikolaos Stagakis received the Electrical Engineering and Computer Technology degree from the University of Patras in 2018. He is currently pursuing the Ph.D. degree with the VVR Laboratory. His current research interests include the design and development of augmented reality systems, 3D graphics, and computer vision.

Evangelia I. Zacharaki received the M.Eng. degree in 1999, the Ph.D. degree in 2004, and the H.D.R. degree (highest accreditation to coordinate research) in informatics from Université Paris-Est in 2017. She was a Post-Doctoral Researcher with the Section of Biomedical Image Analysis, UPenn, USA, from 2005 to 2009; the Medical Physics Department, University of Patras (UPatras), from 2009 to 2012; the Department of Computer Engineering and Informatics, UPatras, from 2012 to 2015 and from 2017 to 2018; the Center for Visual Computing, CentraleSuplec/INRIA, France, from 2015 to 2017; and the Department of Electrical and Computer Engineering, UPatras, from 2018 to 2022. She has been an Assistant Scientist with the Department of Radiation Oncology, University of Miami, USA, since 2022. Her current research interests include machine learning, computational and statistical modeling for representation, and the fusion and analysis of high-dimensional biomedical data, with an emphasis on medical image processing. She received the Marie Curie Reintegration Grant and has participated in more than 16 international (/NIA), European (FP7 and H2020), and national (GSRT and FRM) research projects.

Konstantinos Moustakas (Senior Member, IEEE) received the Diploma and Ph.D. degrees in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 2003 and 2007, respectively. From 2007 to 2011, he was a Post-Doctoral Research Fellow with the Information Technologies Institute, Centre for Research and Technology Hellas. He is currently a Professor with the Electrical and Computer Engineering Department, University of Patras, the Head of the Visualization and Virtual Reality Group, the Director of the Wire Communications and Information Technology Laboratory, and the Director of the M.Sc. Program on Biomedical Engineering, University of Patras. He serves as an Academic Research Fellow with the ISI/Athena Research Center. His current research interests include virtual, augmented, and mixed reality, 3D geometry processing, haptics, virtual physiological human modeling, information visualization, physics-based simulations, computational geometry, and computer vision. During the latest years, he has been the (co)author of more than 250 papers in refereed journals, edited books, and international conferences. He has participated in more than 30 research and development projects funded by the EC and the Greek Secretariat of Research and Technology, while he has served as the coordinator or scientific coordinator in four of them. He has been a member of the organizing committee of several international conferences. He is a Senior Member of the IEEE Computer Society and a member of Eurographics. His research work has received several awards.